Attention Strategies for Multi-Source Sequence-to-Sequence Learning

Jindřich Libovický, Jindřich Helcl

Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
Charles University

August 2, 2017
Introduction

- Attention over multiple source sequences relatively unexplored.
- This work proposes two techniques:
  - *Flat* attention combination
  - *Hierarchical* attention combination
- Applied to tasks of multimodal translation and automatic post-editing.

Motivation

No universal method that models explicitly the importance of each input.
Multi-Source Sequence-to-Sequence Learning

Any number of input sequences with possibly different modalities.

![Multimodal translation example](image)

**Figure 1**: Multimodal translation example.

**Examples**
Multimodal translation, automatic post-editing, multi-source machine translation, ...
In each decoder step $i$

- compute distribution over encoder states given the decoder state
- the decoder gets a context vector to decide about its output

\[
e_{ij} = \mathbf{v}_a^\top \tanh(W_a s_i + U_a h_j)
\] (1)

\[
\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}
\] (2)

\[
c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j
\] (3)
In each decoder step $i$
- compute distribution over encoder states given the decoder state
- the decoder gets a context vector to decide about its output

$$e_{ij} = v_a^\top \tanh(W_a s_i + U_a h_j) \quad (1)$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \quad (2)$$

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \quad (3)$$

What about multiple inputs?
- Widely used technique [Firat et al., 2016, Zoph and Knight, 2016].
- Attention over input sequences computed independently.
- Combination resolved later on in the network
Flat Attention Combination

Importance of different inputs reflected in the joint attention distribution.
Flat Attention Combination

one source → N sources

\[ e_{ij} = v_a^\top \tanh(W_a s_i + U_a h_j) \rightarrow e_{ij}^{(k)} = v_a^\top \tanh(W_a s_i + U_a^{(k)} h_j) \]

\[ \alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \rightarrow \alpha_{ij}^{(k)} = \frac{\exp(e_{ij}^{(k)})}{\sum_{n=1}^{N} \sum_{m=1}^{T_x(n)} \exp(e_{im}^{(n)})} \]

\[ c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \rightarrow c_i = \sum_{k=1}^{N} \sum_{j=1}^{T_x(k)} \alpha_{ij}^{(k)} U_c^{(k)} h_j^{(k)} \]

- \( U_a^{(k)}, U_c^{(k)} \) project states to a common space
- Question: Should \( U_a^{(k)} = U_c^{(k)} \)? (i.e. should the projection parameters be shared?)
Attention distribution is **factored** by input.
Hierarchical Attention Combination

1. Compute the context vector:

\[ c^{(k)}_i = \sum_{j=1}^{T^{(k)}} \alpha_{ij}^{(k)} h^{(k)}_j, \]  
where \( \alpha_{ij}^{(k)} = \ldots \)

...using the vanilla attention

2. Compute another attention distribution over the intermediate context vectors \( c^{(k)}_i \) and get the resulting context vector \( c_i \).

- \( e^{(k)}_i = v^T_b \tanh(W_b s_i + U_b^{(k)} c_i^{(k)}) \)
- \( \beta^{(k)}_i = \frac{\exp(e^{(k)}_i)}{\sum_{n=1}^{N} \exp(e^{(n)}_i)} \)
- \( c_i = \sum_{k=1}^{N} \beta^{(k)}_i U_c^{(k)} c_i^{(k)} \)

- As in the flat scenario, the context vectors have to be projected to a shared space.
- Same question arises – should \( U_b^{(k)} = U_c^{(k)} \)?
Experiments and Results

- Experiments conducted on multimodal translation (MMT) and automatic post-editing (APE)
- In both flat and hierarchical scenarios, we tried both sharing and not sharing the projection matrices.
- Additionally, we tried using the sentinel gate [Lu et al., 2016], which enables the decoder to decide whether or not to attend to any encoder.

Experiments conducted using Neural Monkey, code available here: https://github.com/ufal/neuralmonkey.
## Experiments and Results

<table>
<thead>
<tr>
<th>share</th>
<th>sent.</th>
<th>MMT</th>
<th>APE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>BLEU</td>
<td>METEOR</td>
</tr>
<tr>
<td>concat.</td>
<td></td>
<td>31.4 ± .8</td>
<td>48.0 ± .7</td>
</tr>
<tr>
<td>flat</td>
<td>×</td>
<td>×</td>
<td>30.2 ± .8</td>
</tr>
<tr>
<td></td>
<td>×</td>
<td>✓</td>
<td>29.3 ± .8</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>×</td>
<td>30.9 ± .8</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>29.4 ± .8</td>
</tr>
<tr>
<td>hierarchical</td>
<td>×</td>
<td>×</td>
<td><strong>32.1 ± .8</strong></td>
</tr>
<tr>
<td></td>
<td>×</td>
<td>✓</td>
<td>28.1 ± .8</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>×</td>
<td>26.1 ± .7</td>
</tr>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>22.0 ± .7</td>
</tr>
</tbody>
</table>

Results on the Multi30k dataset and the APE dataset. The column ‘share’ denotes whether the projection matrix is shared for energies and context vector computation, ‘sent.’ indicates whether the sentinel vector has been used or not.
Example

Source:

A man sleeping in a green room on a couch.

Output with attention:

Reference:

ein Mann schläft in einem grünen Raum auf einem Sofa.
• The results show both methods achieve comparable results to the existing approach (concatenation of the context vectors).
• Hierarchical attention combination achieved best results on MMT, and is faster to train.
• Both methods provide a trivial way to inspect the attention distribution w.r.t. the individual inputs.
Conclusions

- The results show both methods achieve comparable results to the existing approach (concatenation of the context vectors).
- Hierarchical attention combination achieved best results on MMT, and is faster to train.
- Both methods provide a trivial way to inspect the attention distribution w.r.t. the individual inputs.

Thank you for your attention!
